

Adaptive Questionnaires for Direct Identification of Optimal Product Design

Max Yi Ren^{*1} and Clayton Scott^{†2}

¹Department of Mechanical Engineering, Arizona State University

²Division of Electrical and Computer Engineering, University of Michigan, Ann Arbor

January 6, 2017

Abstract

We consider the problem of identifying the most profitable product design from a finite set of candidates under unknown consumer preference. A standard approach to this problem follows a two-step strategy: First, estimate the preference of the consumer population, represented as a point in part-worth space, using an adaptive discrete-choice questionnaire. Second, integrate the estimated part-worth vector with engineering feasibility and cost models to determine the optimal design. In this work, we (1) demonstrate that accurate preference estimation is neither necessary nor sufficient for identifying the optimal design, (2) introduce a novel adaptive questionnaire that leverages knowledge about engineering feasibility and manufacturing costs to directly determine the optimal design, and (3) interpret product design in terms of a nonlinear segmentation of part-worth space, and use this interpretation to illuminate the intrinsic difficulty of optimal design in the presence of noisy questionnaire responses. We establish the superiority of the proposed approach using a well-documented optimal product design task. This study demonstrates how the identification of optimal product design can be accelerated by integrating marketing and manufacturing knowledge into the adaptive questionnaire.

^{*}yiren@asu.edu

[†]clayscot@umich.edu

1 Introduction

Understanding consumer preferences is usually necessary during profit-driven product design. A common tool for this cause is Discrete Choice Analysis (DCA). In DCA, the designer collects consumer responses on product alternatives through a questionnaire, and uses the data to create a preference model, which is *subsequently* used for the identification of the optimal (i.e., most profitable) design. While numerous research has followed this two-step procedure (see, e.g., [1–6]), there has been limited theoretical discussion of the relationship between the accuracy of a preference model and the market performance of the resultant product (Some empirical studies can be found in [7, 8]). Although having a precise understanding of consumer preference does not hurt, it may not be necessary for product design: For example, when all but one candidate designs have higher cost than their potential profit, that single design is optimal regardless of our understanding of consumer preference. While this is an extreme case, it suggests that we could potentially accomplish the task of identifying the optimal design with fewer resources spent on the questionnaire if it directly identifies the optimal design without focusing on preference learning as an intermediate step.

This paper aims to turn this intuition into a rigorous questionnaire mechanism. To accomplish this goal, we argue that finding the optimal product is a *group identification* problem (defined below), where the *group label* (the ID of the optimal design) of an *object* (the consumer preference model) is to be identified through binary *queries* (the questionnaire). We further show that an existing algorithm for group identification can be extended to optimal product design.

By drawing this connection, we propose a questionnaire mechanism that adaptively chooses queries to greedily minimize the expected number of queries. We compare this approach with a standard adaptive questionnaire by Abernethy et al. [9] (called Abernethy’s alg. in the sequel) on a well-documented product design problem. The performance of the two methods reflects their different objectives: While Abernethy’s alg. yields better preference estimates, our method has consistently better performance in identifying the optimal product, leading to a significant difference in the expected profit of the resultant product.

The key contributions of this paper are as follows. (1) We clearly explain the mathematical difference between preference modeling and optimal product identification, and thus show that accurate preference estimation is neither necessary nor sufficient for identifying the optimal product design. To summarize, consider the true preference as a point in a “preference

space”: The former is equivalent to estimating the location of that point. In the latter, the entire space is segmented with each segment representing preferences that share the same optimal design. The objective of the latter is thus to identify the true segment, i.e., the segment where the true preference lies. (2) We develop a real-time adaptive questionnaire mechanism that identifies optimal designs more effectively than a standard method in noise-free scenarios, by leveraging engineering feasibility and cost models. (3) We point out a critical issue *by construction* in the use of preference models in product design: Even when the estimation of preference is close to the truth, the model may still lead to an inferior design decision. This happens when the true segment is “thin” or close to the origin, with the latter case representing low sensitivity of consumer preferences to design changes or equivalently, high noise in responses to choice questions. The results presented in this paper call for a more systematic product design methodology when its goal is to serve product designers to identify profitable designs more cost effectively rather than to help marketing researchers and psychologists to precisely understand human preference.

The rest of the paper is structured as follows. Sec. 2 provides preliminary knowledge necessary for in-depth discussion. Sec. 3 introduces the questionnaire mechanism and Sec. 4 its implementation. Sec. 5 presents results from a case study. Lastly, discussions on the limitations of the proposed method and relaxations of its assumptions are presented in Sec. 6.

2 Preliminaries

We introduce background material necessary for an elaboration on this problem statement.

2.1 Profit-driven product design

A profit-driven product design problem can be formulated as follows, with \mathbf{z} being the product design and \mathbf{w} the parameters of a preference model:

$$\max_{\mathbf{z} \in \mathcal{Z}} \text{profit}(\mathbf{z}; \mathbf{w}) := \text{market share}(\mathbf{z}; \mathbf{w}) (\text{price}(\mathbf{z}) - \text{cost}(\mathbf{z})). \quad (1)$$

More specifically, $\mathbf{z} := (z_{(1)}, \dots, z_{(D)}) \in \mathcal{Z}$ is an binary attribute vector where each element represents the existence of a certain attribute level in the product¹. \mathcal{Z} represents the set of all feasible combinations of attribute levels.

¹For example, the attribute “MPG” of a car has levels “20”, “30”, and “40”. The binary encoding of attributes we use in this paper, e.g., $z_{(1)=1} \Leftrightarrow \text{MPG} = 20$, $z_{(2)=1} \Leftrightarrow$

Unit price is indicated by one of the attributes while unit cost is a function of \mathbf{z} often built upon engineering models of the problem. Parameters $\mathbf{w} \in \mathbb{R}^D$ are called *part-worths* of a consumer *utility* model $u(\mathbf{z}; \mathbf{w})$. The preference influences the profit through the market share. Specifically, we use the logit model [10]

$$\text{market share}(\mathbf{z}; \mathbf{w}) \propto \text{sigmoid}(u(\mathbf{z}; \mathbf{w}) - u(\mathbf{z}_0; \mathbf{w})), \quad (2)$$

where \mathbf{z}_0 is some fixed competing product to \mathbf{z} . Extension to multiple competing products is trivial. The RHS of the equation represents the probability of choosing design \mathbf{z} by a consumer, and is proportional to the market share under homogeneous consumer preference, i.e., everyone shares the same deterministic \mathbf{w} . As discussed in Sec. 6, other choice models, e.g., probit and non-compensatory ones [11–13], can replace Eq. (2) without significantly affecting our discussion.

2.2 Preference learning

Clearly, \mathbf{w} plays a critical role in formulating and solving Eq. (1). In reality, however, the true part-worths of a target consumer group, denoted as \mathbf{w}^* , are often unknown. DCA derives a part-worth estimate, denoted as $\hat{\mathbf{w}}$, using a series of queries. Each query consists of a set of design alternatives, from which the participant is asked to pick the relatively more preferred design. The accumulated responses form a data set $\mathcal{S}^{(Q)} := \{(\mathbf{z}_{q_1}, \mathbf{z}_{q_2}, \dots)\}_{q=1}^Q$ where \mathbf{z}_{q_1} is the preferred design in the q th query, and $\{\mathbf{z}_{q_2}, \dots\}$ are the unchosen alternatives. In the sequel, we will assume pairwise choices, and discuss the extension to multiple choices in Sec. 6. When necessary, superscripts will be omitted to avoid clutter. Given this data set and a prior belief of consumer preference $p(\mathbf{w})$, the posterior density function $p(\mathbf{w}; \mathcal{S}^{(Q)})$ can be derived following a regularized logistic or hierarchical Bayes [14, 15] model. Under either model, the maximum a posteriori (MAP) estimator can be found as $\hat{\mathbf{w}} := \arg\max_{\mathbf{w}} p(\mathbf{w}; \mathcal{S})$, and the uncertainty of this point estimate, i.e., its variance-covariance matrix, can be calculated by the inverse of the Hessian $\mathbf{H}(\hat{\mathbf{w}}, \mathcal{S})$ of the negative log-likelihood $-\log(p(\mathbf{w}; \mathcal{S}))$. In this paper, we use a regularized logistic model due to its convexity and real-time computation of MAP estimates. A comparison between these two approaches is discussed

MPG = 30, etc., is a common way to treat nonlinearity in preference with respect to attribute levels.

in [16]. Assuming a regularized logistic model, we have:

$$-\log(p(\mathbf{w}; \mathcal{S})) = \sum_{q'=1}^q \log(1 + \mathbf{w}^T \Delta \mathbf{z}^{(q')}) + \frac{C}{2} \mathbf{w}^T \mathbf{w}, \quad (3)$$

where $\Delta \mathbf{z}^{(q')} := \mathbf{z}_{q'_2} - \mathbf{z}_{q'_1}$, and the parameter C will be selected by cross-validation (see Sec. 4). The goal of preference learning is to find a part-worth estimate close to the ground truth and with low uncertainty (e.g., determinant of $\mathbf{H}(\hat{\mathbf{w}}, \mathcal{S})$), while limiting Q , the number of queries posed.

2.3 Optimal product identification

Since the uncertainty in part-worth estimation always exists under a finite number of queries, one needs to modify Eq. (1) to incorporate $p(\mathbf{w}; \mathcal{S})$. Two modified problem definitions for optimal product identification exist: One is to optimize the expected profit:

$$\max_{\mathbf{z} \in \mathcal{Z}} \mathbb{E}_{\mathbf{w}} [\text{profit}(\mathbf{z}; \mathbf{w})] = \int_{\mathbb{R}^D} \text{profit}(\mathbf{z}; \mathbf{w}) p(\mathbf{w}; \mathcal{S}) d\mathbf{w}, \quad (4)$$

and the second is to optimize the probability of being the most profitable (denoted as $\pi(\mathbf{z})$):

$$\max_{\mathbf{z} \in \mathcal{Z}} \pi(\mathbf{z}) := \mathbb{E}_{\mathbf{w}} [\mathbb{1}(\mathbf{z}; \mathbf{w})] = \int_{\mathbb{R}^D} \mathbb{1}(\mathbf{z}; \mathbf{w}) p(\mathbf{w}; \mathcal{S}) d\mathbf{w}, \quad (5)$$

where $\mathbb{1}(\mathbf{z}; \mathbf{w}) := \mathbb{1}(\text{profit}(\mathbf{z}; \mathbf{w}) > \text{profit}(\mathbf{z}'; \mathbf{w}), \forall \mathbf{z}' \neq \mathbf{z})$ and $\mathbb{1}(\text{condition}) = 1$ when the condition is true and 0 otherwise. The design decision generated from these two objectives may not be consistent for an arbitrary \mathcal{S} . The goal of optimal product identification is to acquire such queries and responses so that the design decision matches the true solution to Eq. (1).

2.4 Assumptions

The majority of the discussion will be based on the following conditions and assumptions: (1) We consider a consumer group who share similar preferences; (2) queries are binary, i.e., each consists of two design alternatives and one must be chosen; (3) \mathcal{Z} is a finite set; (4) the preference model is linear, i.e., $u = \mathbf{w}^T \mathbf{z}$; (5) $\|\mathbf{w}^*\|$ is a large number, reflecting low-noise responses and substantial information for measurement; (6) engineering models, i.e., product costs and feasibility, are deterministic; and (7) the predicted optimal design is derived from Eq. (5). We will discuss in Subsec. 5.5 the issue with response noise; and in Sec. 6 the relaxation of assumptions (1) to (4), as well as the connection between Eqs. (4) and (5).

2.5 Query strategies for preference learning and optimal product identification

A query strategy defines how the q th query shall be created based on the current knowledge $p(\mathbf{w}; \mathcal{S}^{(q-1)})$. For preference learning purposes, existing strategies [9, 16–18] have been developed to greedily minimize the estimation uncertainty of $\hat{\mathbf{w}}$, which is equivalent to maximizing the determinant of $\mathbf{H}(\hat{\mathbf{w}}, \mathcal{S})$ (see Subsec. 3.3 for a discussion of [9]). Below we show that the query strategies for optimal product identification should be different than for preference learning. Without loss of generality, we will use a simple 2D case for visualization.

Consider K candidate products with fixed costs. By setting the true part-worth to some arbitrary \mathbf{w} , we can solve Eq. (1) to derive the optimal product for that particular \mathbf{w} . Doing so for all $\mathbf{w} \in \mathbb{R}^2$ leads to a segmentation of the part-worth space, as illustrated in Fig. 1. Each segment, denoted as \mathcal{W}_k , corresponds to a set of \mathbf{w} that share the same most profitable design \mathbf{z}_k . Due to the nonlinearity in the mapping from preference to profit (see Eqs. (1) and (2)), the segmentation is also nonlinear. To elaborate, any two candidate products \mathbf{z}_1 and \mathbf{z}_2 have the same profit when \mathbf{w} satisfies

$$\frac{\exp(\mathbf{w}^T \mathbf{z}_1)(\text{price}(\mathbf{z}_1) - \text{cost}(\mathbf{z}_1))}{\exp(\mathbf{w}^T \mathbf{z}_1) + \exp(\mathbf{w}^T \mathbf{z}_0)} = \frac{\exp(\mathbf{w}^T \mathbf{z}_2)(\text{price}(\mathbf{z}_2) - \text{cost}(\mathbf{z}_2))}{\exp(\mathbf{w}^T \mathbf{z}_2) + \exp(\mathbf{w}^T \mathbf{z}_0)}. \quad (6)$$

Let the true optimal design, induced by \mathbf{w}^* , be \mathbf{z}_{k^*} , and the solution to Eq. (5) be $\hat{\mathbf{z}}$. Ideally, one would like to obtain a perfect preference model ($\hat{\mathbf{w}} = \mathbf{w}^*$) and thus the correct optimal product ($\hat{\mathbf{z}} = \mathbf{z}_{k^*}$). However, we show in Figs. 1a-b that a good estimation $\hat{\mathbf{w}} \approx \mathbf{w}^*$ does not guarantee $\hat{\mathbf{z}} = \mathbf{z}_{k^*}$, neither does $\hat{\mathbf{z}} = \mathbf{z}_{k^*}$ require $\hat{\mathbf{w}} \approx \mathbf{w}^*$. In addition, readers may notice that the segments cluster around the origin, in which region consumers are less sensitive to design changes². As we will discuss in Subsec. 5.4, this clustering causes difficulty in correctly identifying the optimal design under indifferent preference or noisy responses. It is worth noting, however, that due to the indifference or high noise in consumer preference, all segments near the origin lead to similar expected profit as market shares are more evenly distributed among competing products.

Knowing this, we can further discuss the difference between query strategies for preference learning and for optimal product identification. We start

²Low part-worth values can also be interpreted as high response noise. Salisbury and Feinberg [19] and Louviere et al. [20] pointed out the downsides of mixing these two interpretations. But we do not discuss this issue here.

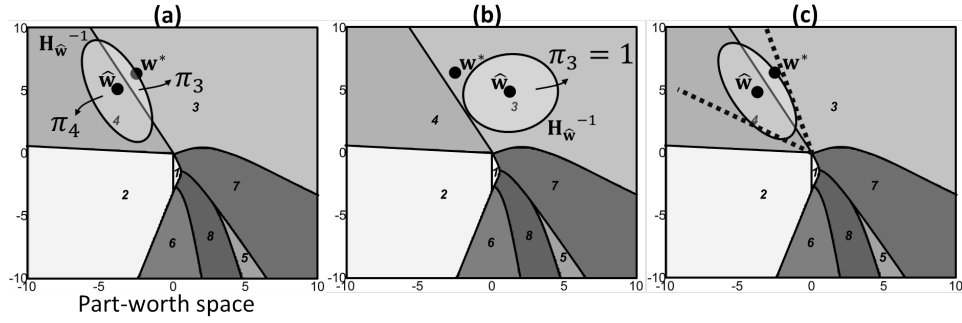


Figure 1: (a) A good estimation $\hat{\mathbf{w}} \approx \mathbf{w}^*$ does not guarantee $\hat{\mathbf{z}} = \mathbf{z}_{k^*}$. π_3 and π_4 are the probabilities for product 3 and 4 to be the optimal, respectively. (b) Achieving $\hat{\mathbf{z}} = \mathbf{z}_{k^*}$ does not necessarily require $\hat{\mathbf{w}} \approx \mathbf{w}^*$. (c) Each query can be considered as a cut in the part-worth space. The posterior $p(\mathbf{w}; \mathcal{S})$ is shifted to the convex cone defined by these cuts. The uncertainty in $\hat{\mathbf{w}}$ is illustrated by the covariance ellipse $\mathbf{H}_{\hat{\mathbf{w}}}^{-1}$. Generation of the figure: In the space of $[-10, 10]^2$, we sample eight points uniformly as candidate products, i.e., each one is represented by two continuous product attributes, one of which is the price. For each part-worth on a dense grid of the space, we calculate \mathbf{z}_{k^*} by Eq. (5). Grid points with the same \mathbf{z}_{k^*} share the same color.

from a prior probability density $p(\mathbf{w})$ (e.g., a multivariate normal distribution or a distribution that leads to maximum entropy, see the entropy definition in Subsec. 3.1). When a user response $(\mathbf{z}_{q_1}, \mathbf{z}_{q_2})$ is collected from iteration q , we update $p(\mathbf{w}; \mathcal{S}^{(q-1)})$ so that the probability mass is shifted towards the half space $\{\mathbf{w} | \mathbf{w}^T (\mathbf{z}_{q_1} - \mathbf{z}_{q_2}) > 0\}$. After some Q responses are collected, $p(\mathbf{w} | \mathcal{S}^{(Q)})$ will concentrate in a cone that contains \mathbf{w}^* (Fig. 1c). A good query strategy for learning \mathbf{w}^* is thus to quickly “narrow” this cone. The algorithms proposed by Toubia et al. [17,18], Abernethy et al. [9], Tong and Koller [21], and Jamieson and Nowak [22] all follow the idea of narrowing the feasible part-worth space by halving it.

For optimal product identification, on the other hand, \mathbf{z}_{k^*} is correctly identified when $p(\mathbf{w}; \mathcal{S}^{(Q)})$ concentrates in the segment \mathcal{W}_{k^*} , i.e., when $p(\mathbf{w}; \mathcal{S}) > 0$ only in \mathcal{W}_k , $\hat{\mathbf{z}} = \mathbf{z}_{k^*}$ according to both Eq. (4) and Eq. (5). Therefore, a good query strategy in this case should be to quickly identify the correct “group” \mathcal{W}_k , by shifting the mass of $p(\mathbf{w}; \mathcal{S}^{(Q)})$ towards it, rather than towards the point \mathbf{w}^* . Jamieson and Nowak [22] shares a similar problem to ours in that a particular segment rather than a point is to be identified from a segmentation of a space. However, in [22], the segments are created by hyperplanes directly defined by pair-wise comparisons, while in this paper, the nonlinear segments are induced from the candidate products, independent from the hyperplanes defined by the collection of queries. See Fig. 2 for a comparison.

3 Query Strategy

This section addresses the central question of how queries should be adaptively made to directly minimize the expected number of queries needed to identify the optimal design. To do so, we show the connection between this problem and the problem of “group identification”, and introduce an extension to the Group Identification Splitting Algorithm [23] that solves our problem. We also review an existing strategy for preference learning from Abernethy et al. [9] (called Abernethy’s alg. in the sequel) that will be used as a benchmark in the case study. Implementation details of GISA and Abernethy’s alg. will be discussed in Sec. 4.

3.1 Group identification and optimal product identification

Group identification is an extension of *object identification*, also known as the twenty-questions game. Consider a game between players A and B. Both players are familiar with a list of objects and the group labels they

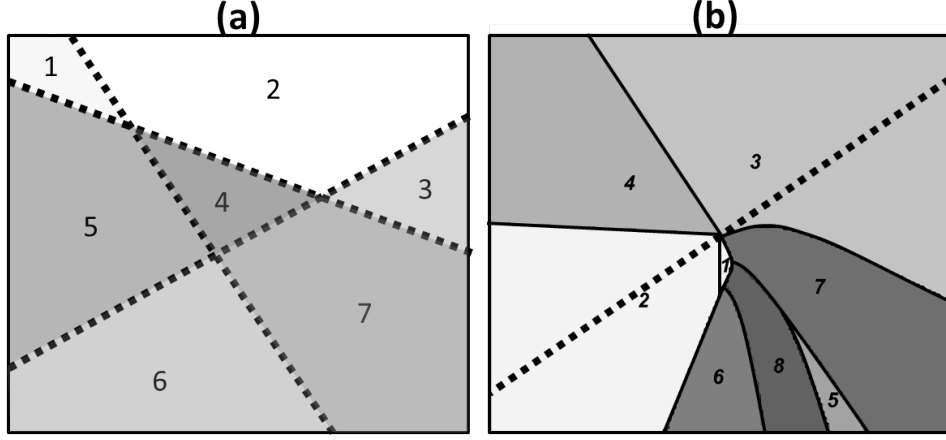


Figure 2: Comparison on problem settings between (a) [22] and (b) this paper: (a) Each segment (shaded polygons) represents a preference ranking and each hyperplane a pair-wise comparison query (dotted lines) (b) The nonlinear segments (shaded) are induced by the candidate products and their costs, while queries (dotted line) are pair-wise comparisons

belong to, see Table 1. Player A first picks an object from the list randomly. In object identification, Player B needs to identify the object selected by Player A, whereas in group identification, Player B only needs to determine the group to which the object belongs. To know which group Player A picked from, Player B can pick queries in the columns to ask. Based on Player A’s responses, which are binary, Player B either picks new queries or makes a guess on the group label. Player B’s query strategy affects how quickly he can identify the correct group. In this example, an intuitively good query to start with is “Q1”, since its answer directly determines which group the object is from, i.e., it allows one of the group labels to have a probability of one conditioned on the response. It is worth noting that we cannot consider groups as “meta-objects”, since objects in the same group can lead to different query responses.

We now show that optimal product identification in our context shares the same elements as in Table 1. Consider the set of objects to be all part-worth vectors in \mathbb{R}^D . Each induces a list of binary responses to pairwise queries, which are defined by \mathcal{Z} . Specifically, queries are determined by $\text{sign}\{\mathbf{w}^T(\mathbf{z}_i - \mathbf{z}_j)\}$ for all pairs $\mathbf{z}_i, \mathbf{z}_j$. A group is a set \mathcal{W}_k of part-worth vectors that share the same most profitable design. Similar to Player B who

Table 1: An example of the group identification problem

Group	Heroes		Villains	
Object	Superman	Batman	Catwoman	Joker
Prior Prob.	0.25	0.25	0.25	0.25
Q1:Mask?	Yes	Yes	No	No
Q2:Can fly?	Yes	No	No	No
Q3:Female?	No	No	Yes	No

has a (non-informative) prior knowledge of which object Player A picks, here objects are associated with a prior probability density $p(\mathbf{w})$, and each group has a prior probability mass $\pi_k^{(0)} = \int_{\mathcal{W}_k} p(\mathbf{w}) d\mathbf{w}$. Upon acquiring the q th user response, we can update $\mathcal{S}^{(q)}$, the posterior $p(\mathbf{w}; \mathcal{S}^{(q)})$ and

$$\pi_k^{(q)} = \int_{\mathcal{W}_k} p(\mathbf{w}; \mathcal{S}^{(q)}) d\mathbf{w}. \quad (7)$$

Since $p(\mathbf{w}; \mathcal{S}^{(q)})$ is a probability density function, we have $\sum_{k=1}^K \pi_k^{(q)} = 1$. One can see that both group identification and optimal product identification share the same process of converging $\pi_{k^*}^{(Q)}$ to 1 by collecting responses to binary queries. The differences from the latter are that (1) we have an infinite number of objects (candidate part-worth vectors), and (2) the conditional probabilities are not proportional to the prior. For example, in Table 1, the conditional probabilities for “superman” and “batman” are still equal after receiving “yes” from Q1; while in the case of optimal product identification, each response will nonlinearly affect $p(\mathbf{w}; \mathcal{S}^{(q)})$, which in turn determines $\Pi^{(q)} := (\pi_1^{(q)}, \dots, \pi_K^{(q)})$.

3.2 The Group Identification Splitting Algorithm (GISA)

GISA greedily minimizes the expected number of queries needed to reach $H(\Pi^{(Q)}) = 0$, which occurs iff all probability mass is on one design. Here we explain the algorithm in the context of optimal product identification. Readers are referred to [23] for the derivation of the algorithm for group identification. Let a “path” be a sequence of queries and responses. Upon receiving the response for each query, we move from one node on the path to the next, and update $\Pi^{(q)}$. With enough queries, we will reach $\pi_{k^*}^{(Q)} = 1$. Given a query strategy, one can generate multiple paths for objects (\mathbf{ws}) drawn from the prior ($p(\mathbf{w})$). These paths form a binary decision tree (Fig. 3.2). Note that multiple leaf nodes of the tree can share the same group label. The expected number of queries needed to reach a leaf, denoted as

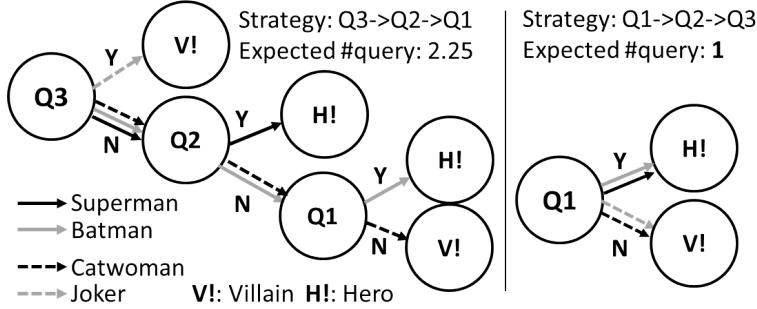


Figure 3: Binary decision trees under two query strategies.

L , can be calculated as a function of $p(\mathbf{w})$ and the query strategy. While optimizing L with respect to all possible query strategies is shown to be NP complete [24], Bellala et al. showed that a greedy approach to solve the problem is available as L can be decomposed into a set of additive terms $\tilde{L}^{(a)}$ with respect to each internal node a of the binary tree [23]. Therefore, one greedy strategy is to minimize $\tilde{L}^{(q)}$ from node 1 to Q along the actual path taken during a questionnaire. The local objective $\tilde{L}^{(q)}$ takes the following form:

$$\tilde{L}^{(q)} = 1 - H(\rho^{(q)}) + \sum_{k=1}^K \pi_k^{(q)} H(\rho_k^{(q)}), \quad (8)$$

where $H(\pi) := -\pi \log_2 \pi - (1 - \pi) \log_2 (1 - \pi)$ is the binary entropy of scalar π , and $0 \log_2 0 := 0$. The “reduction factor” $\rho^{(q)}$ and the “group reduction factor” $\rho_k^{(q)}$ depend on the candidate query, and are defined as follows. For a binary query based on \mathbf{z}_i and \mathbf{z}_j , the two child nodes, denoted as l (for left) and r (for right), are induced by responses $(\mathbf{z}_i, \mathbf{z}_j)$ and $(\mathbf{z}_j, \mathbf{z}_i)$, respectively. We define

$$\pi_l^{(q)} = \int \mathbb{1}(\mathbf{w}^T(\mathbf{z}_i - \mathbf{z}_j) > 0) p(\mathbf{w}; \mathcal{S}^{(q-1)}) d\mathbf{w} \quad (9)$$

and $\pi_r^{(q)} = 1 - \pi_l^{(q)}$. These are the conditional probabilities of branching left and right, respectively. $\rho^{(q)}$ is defined as

$$\rho^{(q)} = \max\{\pi_l^{(q)}, \pi_r^{(q)}\}. \quad (10)$$

Similarly, for each design k we define

$$\pi_{l,k}^{(q)} = \int_{\mathcal{W}_k} \mathbb{1}(\mathbf{w}^T(\mathbf{z}_i - \mathbf{z}_j) > 0) p(\mathbf{w}; \mathcal{S}^{(q-1)}) d\mathbf{w}, \quad (11)$$

and $\pi_{r,k}^{(q)} = \pi_k^{(q)} - \pi_{l,k}^{(q)}$. These are the probabilities of branching left and right, respectively, for part-worths that make \mathbf{z}_k the optimal. $\rho_k^{(q)}$ is defined as

$$\rho_k^{(q)} = \max\{\pi_{l,k}^{(q)}, \pi_{r,k}^{(q)}\} / \pi_k^{(q)}. \quad (12)$$

GISA picks from the candidate query set \mathcal{Q} a new query that minimizes $\tilde{L}^{(q)}$. An ideal query would achieve $\rho^{(q)} = 0.5$ and $\rho_k^{(q)} = 0$ or 1 , i.e., the query “halves” the part-worth space (by $\rho^{(q)} = 0.5$) but preserves entire groups (by $\rho_k^{(q)} = 0$ or 1). In practice, however, the two parts of the objective often trade off. In contrast, algorithms for preference estimation are only concerned with the former criterion. We also terminate the questionnaire after a pre-determined number Q of queries rather than when $H(\Pi^{(q)}) = 0$. It should be noted that in the original presentation of GISA [23], the space of objects was discrete and thus calculation of $\pi_k^{(q)}$, $\pi_l^{(q)}$, and $\pi_{l,k}^{(q)}$ was straightforward. Our extension to optimal product identification is thus nontrivial as we must develop computationally efficient methods for calculating these probabilities. Alg. 1 provides a sketch of the algorithm. Implementation details will be discussed in Subsec. 4.1.

Algorithm 1 Group Identification Splitting Algorithm

- 1: **Input:** \mathcal{Z} , $p(\mathbf{w})$, $q = 0$, Q , unit cost, u_0
 - 2: **Output:** $\Pi^{(Q)}$
 - 3: **while** $q \leq Q$ **do**
 - 4: calculate $\Pi^{(q)}$ (Eq. (7))
 - 5: calculate $\rho^{(q)}$ and $\rho_k^{(q)}$ (Eqs. (10), (12))
 - 6: calculate \tilde{L}_q (Eq. (8)) for each query in \mathcal{Q}
 - 7: choose the next query with the minimal \tilde{L}_q
 - 8: collect query response, set $q = q + 1$, update $\mathcal{S}^{(q)}$ and $p(\mathbf{w}; \mathcal{S}^{(q)})$
-

To demonstrate the algorithm, let us revisit the group identification problem from Table 1. For Player B to pick a query, he will first specify prior probability masses for each object to be the one Player A chooses. Without any knowledge, he assumes that the four objects have equal chances, leading to $\Pi = (\pi_{\text{Heroes}}, \pi_{\text{Villains}}) = (0.5, 0.5)$. The calculation of \tilde{L} follows Eqs. (8), (10) and (12) and is summarized in Table 2, where l and r represent child nodes for responses “yes” and “no”, respectively. From the table, “Q1” has the minimum \tilde{L} and thus will be chosen. It should be noted that the choice of query depends on the prior knowledge. For example, if it is known that Player A will never choose “Joker”, then Q1 and Q2 are equally plausible.

Table 2: Query selection for the case in Table 1

Query	Q1	Q2	Q3
$\pi_l (\pi_r)$	0.5 (0.5)	0.25 (0.75)	0.25 (0.75)
ρ	0.5	0.75	0.75
$\pi_{l,(\text{Heroes})} (\pi_{r,(\text{Heroes})})$	1 (0)	0.5 (0.5)	0 (1)
$\pi_{l,(\text{Villains})} (\pi_{r,(\text{Villains})})$	0 (1)	0 (1)	0.5 (0.5)
$\rho(\text{Heroes})$	1	0.5	1
$\rho(\text{Villains})$	1	1	0.5
\tilde{L}	0.5	1.16	1.16

3.3 Abernethy’s algorithm

Abernethy’s alg. chooses a new query based on two criteria: (1) “utility balance” requires the query to have the smallest utility gap $|(\mathbf{z}_i - \mathbf{z}_j)^T \hat{\mathbf{w}}|$ where $\hat{\mathbf{w}}$ is the current part-worth estimator; and (2) “minimax variance” requires the query to minimize the largest estimation variance. For the latter criterion, we calculate the Hessian of the negative log likelihood $\mathbf{H}(\hat{\mathbf{w}}, \mathcal{S}^{(q)}) := \frac{\partial^2 (-\log p(\mathbf{w}; \mathcal{S}^{(q)}))}{\partial \mathbf{w}^2} \Big|_{\mathbf{w}=\hat{\mathbf{w}}}$:

$$\mathbf{H}(\hat{\mathbf{w}}, \mathcal{S}^{(q)}) = \frac{1}{C} \mathbf{I} + \sum_{q'=1}^q \frac{\exp(-\hat{\mathbf{w}}^T \Delta \mathbf{z}^{(q')})}{(1 + \exp(-\hat{\mathbf{w}}^T \Delta \mathbf{z}^{(q')}))^2} \Delta \mathbf{z}^{(q')} \Delta \mathbf{z}^{(q')T}, \quad (13)$$

and its projection to the subspace perpendicular to $\hat{\mathbf{w}}$:

$$\mathbf{H}_{\hat{\mathbf{w}}} = \left(\mathbf{I} - \frac{\hat{\mathbf{w}} \hat{\mathbf{w}}^T}{\|\hat{\mathbf{w}}\|^2} \right) \mathbf{H}(\hat{\mathbf{w}}, \mathcal{S}^{(q)}). \quad (14)$$

The largest estimation variance of $\hat{\mathbf{w}}$ (in the subspace perpendicular to $\hat{\mathbf{w}}$) corresponds to the eigenvector \mathbf{v} of the smallest eigenvalue of $\mathbf{H}_{\hat{\mathbf{w}}}$. The projection along $\hat{\mathbf{w}}$ is used to ensure that $\mathbf{v}^T \hat{\mathbf{w}} = 0$, which satisfies “utility balance”. Ideally, \mathbf{v} satisfies both criteria, if there exist \mathbf{z}_i and \mathbf{z}_j such that $\mathbf{v} = \mathbf{z}_i - \mathbf{z}_j$. In practice, we choose a query so that $\mathbf{z}_i - \mathbf{z}_j$ is close to \mathbf{v} and perpendicular to $\hat{\mathbf{w}}$. Alg. 2 provides a sketch of this algorithm, and implementation details are deferred to Subsec. 4.2.

4 Algorithm Implementation

Some implementation details of GISA and Abernethy’s alg. are elaborated in this section.

Algorithm 2 Abernethy’s alg.

- 1: **Input:** set \mathcal{Z} , $p(\mathbf{w})$, $q = 0$, Q
 - 2: **Output:** $\Pi^{(q)}$, $\hat{\mathbf{w}}$
 - 3: **while** $q \leq Q$ **do**
 - 4: calculate $\mathbf{H}_{\hat{\mathbf{w}}}$ (Eq. (13))
 - 5: find the eigenvector \mathbf{v} of the smallest eigenvalue of $\mathbf{H}_{\hat{\mathbf{w}}}$
 - 6: find $\mathbf{z}_{i,j} = \operatorname{argmin}_{\mathcal{Q}} |\mathbf{v}^T(\mathbf{z}_i - \mathbf{z}_j)|$
 - 7: collect query response, set $q = q + 1$, update $\mathcal{S}^{(q)}$ and $p(\mathbf{w}; \mathcal{S}^{(q)})$
-

4.1 Implementation of GISA

Preference learning Following Eq. (3), the posterior $p(\mathbf{w}; \mathcal{S})$ has the following form:

$$p(\mathbf{w}; \mathcal{S}^{(q)}) \propto \exp\left(-\frac{\mathbf{w}^T \mathbf{w}}{2C}\right) \prod_{q'=1}^q \frac{1}{1 + \exp(\mathbf{w}^T \Delta \mathbf{z}^{(q')})}. \quad (15)$$

The maximum a posteriori (MAP) estimator is derived using the standard *liblinear* package [25] with a ten-fold cross-validation for $C \in \{0.1, 1, 10, \dots, 10^8\}$ after each query. Once C is determined, the resultant MAP estimator will be used to initialize the following MCMC procedure.

Numerical integration through Markov chain Monte Carlo (MCMC)

Recall that executing GISA involves calculating π_l , $\pi_{l,k}$ and π_k by integrating with respect to the density $p(\mathbf{w}; \mathcal{S})$. Here we discuss a MCMC approach to calculating these integrals in the absence of their analytical forms. We noticed during the development that the performance of GISA can be sensitive to the accuracy of these numerical integrals, and illustrate this issue using an artificial 2D case: Consider the three segments \mathcal{W}_1 , \mathcal{W}_2 , \mathcal{W}_3 in Fig. 4a, among which \mathcal{W}_2 contains \mathbf{w}^* and is “narrow”. In this case, a good estimator $\hat{\mathbf{w}}$ can still fall into \mathcal{W}_1 or \mathcal{W}_3 . With a coarse approximation, we could have $\pi_2^{(q)} = 0$, and thus miss the true optimal design. Addressing this issue is non-trivial, particularly when responses are close to noise-free. To elaborate, Fig. 4b shows an unnormalized density $p(\mathbf{w}; \mathcal{S})$ in a 2D case where \mathcal{S} is composed of two vectors $[1, -0.6]$ and $[-0.6, 1]$. The function forms a plateau between the two vectors, and only drops in value far away from the origin due to the prior $p(\mathbf{w})$. Therefore, an accurate numerical integration requires samples to be drawn from the entire plateau. More generally, when responses are noise-free, \mathcal{S} defines a convex cone $\{\mathbf{w} \in \mathbb{R}^D : \Delta \mathbf{Z} \mathbf{w} \geq \mathbf{0}\}$ as

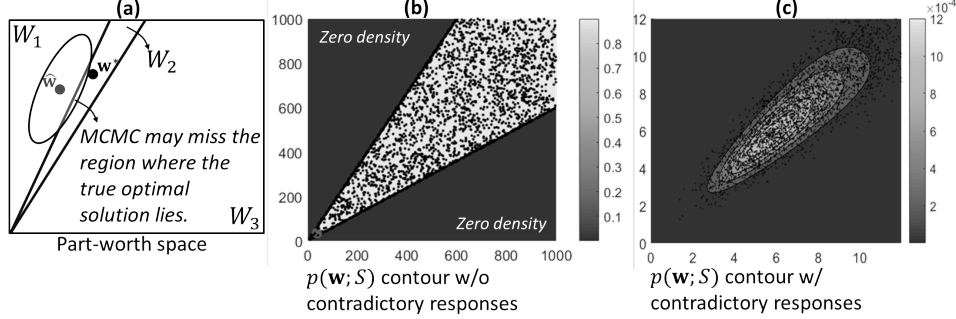


Figure 4: (a) When \mathcal{W}_{k^*} is narrow, naive MCMC has difficulty in identifying the true optimal design due to coarse numerical integrals. (b) Contour of $p(\mathbf{w}; \mathcal{S})$ in 2D. When responses are noise-free, the posterior distribution of \mathbf{w} forms a plateau in the cone defined by $\Delta \mathbf{Z}$. (c) Contour of $p(\mathbf{w}; \mathcal{S})$ when contradictory responses exist. Generation of the contour: $\Delta \mathbf{Z}$ is composed of 10 rows of both $[1, -0.6]$ and $[-0.6, 1]$, and one row of $[-1, 0.6]$ and $[0.6, -1]$. The latter two represent choices against the true preference due to response noise.

the support of $p(\mathbf{w}; \mathcal{S})$, where the q th row of $\Delta \mathbf{Z}$ represents the difference $\Delta \mathbf{z}^{(q)} = \mathbf{z}^{(q,1)} - \mathbf{z}^{(q,2)}$. With a large number of queries, the cone becomes narrow, in which case both the standard Metropolis Hastings (MH) [26] or adaptive Metropolis [27] algorithms produce a low acceptance rate, thus causing the aforementioned coarse approximation. On the other hand, increasing the response noise level would cause contradictory responses to be observed more often, leading to a more smoothed density as can be seen in Fig. 4c. The adaptive Metropolis algorithm can be applied in this case.

To this end, we introduce the following MH procedure that is applicable to all noise levels: Alg. 3 is modified from a standard MH for convex cones [28]. The latter produces a sequence of samples, each of which perturbs from the current sample \mathbf{w}_{old} by $\delta \mathbf{d}$, where \mathbf{d} is a random unit vector, and δ is drawn uniformly from $[\delta_l, \delta_u]$ to ensure that the new sample, if accepted, is within the cone. Since the proposal density is a constant $((\delta_u - \delta_l)^{-1})$, the acceptance level is simply $\alpha = \min \left\{ \frac{p(\mathbf{w}; \mathcal{S})}{p(\mathbf{w}_{\text{old}}; \mathcal{S})}, 1 \right\}$. Note that the cone is only guaranteed to be non-empty when responses are noise-free. To incorporate response noises, i.e., rows of $\Delta \mathbf{Z}$ with opposite signs, these rows are removed before running MH for the remaining convex cone. When the remaining cone is empty, the adaptive Metropolis algorithm will be applied.

Algorithm 3 Metropolis Hastings with a tailored proposal

- 1: **Input:** $\Delta\mathbf{Z}$, $p(\mathbf{w}; \mathcal{S})$, $\hat{\mathbf{w}}$, sample size J
 - 2: **Output:** MH sample set \mathcal{W}_{MH}
 - 3: set $\mathcal{W}_{\text{MH}} = \emptyset$, $\mathbf{w}_{\text{old}} = \hat{\mathbf{w}}$
 - 4: delete rows of $\Delta\mathbf{Z}$ that have opposite signs, i.e., if $\Delta\mathbf{z}^{(q_1)} = -\Delta\mathbf{z}^{(q_2)}$ for some queries q_1 and q_2 , these hyperplanes will be removed
 - 5: **if** the remaining $\Delta\mathbf{Z}$ is not empty **then**
 - 6: **while** $j \leq 2J$ **do**
 - 7: draw \mathbf{d} uniformly from the unit sphere in \mathbb{R}^D
 - 8: calculate $\mathbf{a} := \Delta\mathbf{Z}\mathbf{d}$ and $\mathbf{b} := -\Delta\mathbf{Z}\mathbf{w}_{\text{old}}$
 - 9: define \mathbf{a}_+ and \mathbf{b}_+ (\mathbf{a}_- and \mathbf{b}_-) as a sub-vector of \mathbf{a} and \mathbf{b} , respectively, where elements of \mathbf{a} are positive (negative)
 - 10: define $\delta_l := \max(\mathbf{b}_+/\mathbf{a}_+)$ and $\delta_u := \min(\mathbf{b}_-/\mathbf{a}_-)$, where $/$ is element-wise division
 - 11: if δ_l (δ_u) does not exist, set $\delta_l = -10^3$ ($\delta_u = 10^3$)
 - 12: draw δ uniformly from $[\delta_l, \delta_u]$
 - 13: $\mathbf{w} = \mathbf{w}_{\text{old}} + \delta\mathbf{d}$
 - 14: calculate $\alpha = \min \left\{ \frac{p(\mathbf{w}; \mathcal{S})}{p(\mathbf{w}_{\text{old}}; \mathcal{S})}, 1 \right\}$
 - 15: draw u uniformly from $[0, 1]$
 - 16: **if** $\alpha > u$ **then**
 - 17: $\mathcal{W}_{\text{MH}} \leftarrow \mathbf{w}$, $\mathbf{w}_{\text{old}} = \mathbf{w}$
 - 18: **else**
 - 19: $\mathcal{W}_{\text{MH}} \leftarrow \mathbf{w}_{\text{old}}$
 - 20: $j = j + 1$
 - 21: **else**
 - 22: perform adaptive Metropolis according to [27]
 - 23: return the second half of \mathcal{W}_{MH}
-

In addition, the algorithm uses a single chain, and discards the first half of samples as burn-in samples. We do not adjust the algorithm for a target acceptance rate, unless adaptive Metropolis is used, when the target rate is 0.255. Figs. 4b,c demonstrate the performance of Alg. 3 under noise-free and noisy settings, respectively. Note that the scale of the contours in Figs. 4b,c are less important since the density functions are unnormalized.

Choice of query candidates In practice, the candidate query size $|Q|$ can be large as it grows quadratically with respect to $|\mathcal{Z}|$. A large $|Q|$ can make query selection costly. To this end, we propose to select a small set of N candidate queries in each iteration using two heuristic criteria: Half of the candidates are chosen by first sorting designs with respect to Π in descending order, and pairing designs starting from the top of this sorted list. E.g., let $\pi_{i_1} > \pi_{i_2} > \pi_{i_3}$. Then the candidate pairs to be queried are generated in the order of \mathbf{z}_{i_1} and \mathbf{z}_{i_2} , \mathbf{z}_{i_1} and \mathbf{z}_{i_3} , and lastly \mathbf{z}_{i_2} and \mathbf{z}_{i_3} . The second half are chosen based on Abernethy’s “minimax variance” criterion (see Subsec. 4.2). Qualitatively, these criteria allow us to pick from queries that are either formed by highly probable designs, or those that will help to reduce the variance of $\hat{\mathbf{w}}$. In addition, we omit the queries already chosen for the questionnaire.

4.2 Implementation of Abernethy’s algorithm.

Choice of query candidates In each iteration, $\hat{\mathbf{w}}$ and $\mathbf{H}(\hat{\mathbf{w}}, \mathcal{S}^{(q)})$ are updated in the same way as in GISA³. The eigenvector \mathbf{v} is calculated based on $\mathbf{H}_{\hat{\mathbf{w}}}$ (see Subsec. 3.3). To find such a query where $(\mathbf{z}_i - \mathbf{z}_j)$ is close to \mathbf{v} and perpendicular to $\hat{\mathbf{w}}$, we calculate $\mathbf{c}_1 = |\Delta\mathbf{Z}\hat{\mathbf{w}}|$, where $\Delta\mathbf{Z}$ is a $|Q|$ -by- d matrix, each row of which represents the difference between two designs of a candidate query in Q ; and $\mathbf{c}_2 = |\Delta\mathbf{Z}\mathbf{v}|/||\Delta\mathbf{Z}||$, where $||\Delta\mathbf{Z}||$ is a $|Q|$ -by-1 vector with each element being the Euclidean norm of the corresponding row of $\Delta\mathbf{Z}$, and the division is element-wise. The query is then chosen by the following procedure: We first choose queries with the highest value in \mathbf{c}_2 . If the resulting set contains more than one query, we choose a subset

³Note that Abernethy’s original algorithm uses a regularization network (with squared loss and l^2 regularization on the estimates) for analytical part-worth estimation and thus has a different form of Hessian. It should be noted that due to the usage of a squared penalty (which allows the analytical solution), this formulation (see [9] for details) could falsely penalize an estimator that is consistent with query responses. The regularized logistic problem in Eq. (15), on the other hand, does not have this issue and its MAP estimate can be derived by *liblinear* with a complexity of $O(Q)$, provided that the preference model is linear wrt \mathbf{w} . Thus we choose the latter formulation for estimation.

with the minimum value in \mathbf{c}_1 . If multiple queries still remains, we pick the first one in the list. Similar to GISA, we remove previously chosen queries from \mathcal{Q} during the questionnaire.

5 Case study

We introduce a dial-readout scale design problem from Michalek et al. [2] to demonstrate the difference between GISA and Abernethy’s alg..

5.1 Design attributes and preference part-worths

This problem considers six attributes that affect consumer preference: weight capacity, aspect ratio, i.e., platform length divided by width, platform area, tick mark gap, i.e., distance between 1-lb tick marks, size of readout number and price. Discrete levels of these attributes are listed in Table 3. Thus

Table 3: Design Attributes and Price Levels [2]

Description	Units	Levels				
Weight Capacity	lbs	200	250	300	350	400
Aspect Ratio	-	6/8	7/8	8/8	8/7	8/6
Platform Area	in. ²	100	110	120	130	140
Tick Mark Gap	in.	2/32	3/32	4/32	5/32	6/32
Number Size	in.	0.75	1.00	1.25	1.50	1.75
Price	\$	10	15	20	25	30

each design can be encoded by a 24 dimensional binary vector (Subsec. 5.3 explains why it is not 30 dimensional). The “true” part-worths, denoted as \mathbf{w}^* , were derived through a CBC study in [2] and are presented in Table 4.

5.2 Candidate design set

The engineering model from [2] maps some fourteen design variables, denoted as \mathbf{x} , to the attributes⁴, and specifies constraints on the variables. Specifically, let $\mathbf{z}(\mathbf{x})$ be the mapping from design variables \mathbf{x} to attributes \mathbf{z} , and $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$ be the engineering constraints. To find the set of feasible

⁴“Variables” are parameters that the designer has control of (e.g., engine size), while “attributes” are performance measures resulting from the variables, e.g., acceleration time of the car.

Table 4: True part-worth values [2]

Weight Capacity		Aspect Ratio		Platform Area	
200 lbs.	-0.534	0.75 in.	-0.744	100 in. ²	0.015
250 lbs.	0.129	1.00 in.	-0.198	110 in. ²	-0.098
300 lbs.	0.228	1.25 in.	0.235	120 in. ²	0.049
350 lbs.	0.104	1.50 in.	0.291	130 in. ²	0.047
400 lbs.	0.052	1.75 in.	0.396	140 in. ²	-0.033
Tick Mark Gap		Number Size		Price	
2/32 in.	-0.366	0.75	-0.058	\$10	0.719
3/32 in.	-0.164	0.88	0.253	\$15	0.482
4/32 in.	0.215	1.00	0.278	\$20	0.054
5/32 in.	0.194	1.14	-0.025	\$25	-0.368
6/32 in.	0.100	1.33	-0.467	\$30	-0.908

designs in the attribute space, we solve the following problem:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \|\mathbf{z} - \mathbf{z}(\mathbf{x})\|_2 \\ \text{subject to} \quad & \mathbf{g}(\mathbf{x}) \leq \mathbf{0}. \end{aligned} \tag{16}$$

A design \mathbf{z} is considered feasible if a solution \mathbf{x} is found and the resulting minimal discrepancy $\|\mathbf{z} - \mathbf{z}(\mathbf{x})\|_2$ is less than a tolerance of 10^{-3} . This procedure creates a set \mathcal{Z} with $K = 2455$ feasible designs, $5^6 = 15625$ out of the possible combinations.

5.3 Simulation setup

For both GISA and Abernethy’s alg., the simulated questionnaire is set up as follows. We initialize by calculating the true optimal design \mathbf{z}_{k^*} (Eq. (1)) using $\theta \mathbf{w}^*$. We set $\theta = 100$ to simulate noise-free responses: In this case, the false response rate $\text{sigmoid}(\theta \Delta u)$, for a utility gap $\Delta u := \mathbf{w}^T(\mathbf{z}_i - \mathbf{z}_j)$ between two designs, is less than 0.01% when $\Delta u = 0.1$, i.e., consumers have a chance of 0.01% to choose a product with less utility from a pair when the utility difference between the two is 0.1.

Note that part-worth estimation may encounter an identifiability issue: For any $\hat{\mathbf{w}}$ that optimizes Eq. (15) with a large enough C , in which case the regularization term can be neglected, an arbitrary part-worth vector $\hat{\mathbf{w}} + \delta_{\mathbf{w}}$ is also a solution, provided that $\delta_{\mathbf{w}}$ has the same part-worth values for attribute levels of each attribute. While the regularization term can theoretically address this issue when C is small, the value of C is typically

large in the noise-free case. This is because through cross-validation, the algorithm learns to fully trust the responses rather than the prior. Therefore, we constrain $\hat{\mathbf{w}}$ by forcing the part-worth of the 5th level of each attribute, and thus $\delta\mathbf{w}$, to be zero. This treatment requires a slightly modification to the true part-worths \mathbf{w}^* from Table 4, by also shifting part-worth values of each attribute so that value of the 5th level of each attribute is zero⁵. This treatment leaves the part-worth space to be \mathbb{R}^{24} .

The candidate query set \mathcal{Q} is set to contain all design pairs from \mathcal{Z} with $2455(2455 - 1)/2 = 3012285$ queries. The first query of the questionnaire is randomly picked from \mathcal{Q} . For a query with designs \mathbf{z}_i and \mathbf{z}_j , the simulated user chooses \mathbf{z}_i with probability sigmoid $(\theta(\mathbf{z}_i - \mathbf{z}_j)^T \mathbf{w}^*)$. The lumped preference u_0 (for calculating the market share) is set by uniformly picking a competing product from \mathcal{Z} during the initialization. To compare the performance of GISA and Abernethy’s, we conduct $T = 20$ questionnaires using both algorithms, each consisting of $Q = 100$ queries. For GISA, we set the MCMC sample size to $J = 10^3$ and the size of the heuristic candidate set to be $N = 100$.

5.4 Results and analysis

Performance measures To compare the performance of GISA and Abernethy’s alg., four measures are tracked along the questionnaire: (1) The probability $\pi_{k^*}^{(q)}$ of the true optimal design being identified as optimal, (2) the percentage of correct guesses across all independent runs: $\text{correct\%} := \frac{1}{T} \sum_{t=1}^T \mathbb{1}(\pi_{k^*}^{(q)} > \pi_k^{(q)}, \forall k \neq k^*)$, (3) the correlation $\hat{c} = \text{corr}(\hat{\mathbf{w}}, \mathbf{w}^*)$, and (4) the estimation error $\hat{d} = \|\hat{\mathbf{w}} - \mathbf{w}^*\|$. An algorithm is more effective at identifying the true optimal design if it has higher $\pi_{k^*}^{(q)}$ and correct\% along the progress. On the other hand, the algorithm is better at identifying the true part-worths if \hat{d} and \hat{c} converge faster.

Main findings Results from the case study are presented in Fig. 5: Under the noise-free condition, GISA outperforms Abernethy’s alg. at identifying the optimal design while Abernethy’s has better part-worth estimation with small Q .

⁵Another common practice of addressing the identifiability issue is to constrain part-worths for each attribute to a constant, e.g., 0. However, doing so will introduce equality constraints to the estimation problem which are not naturally supported by the *liblinear* solver.

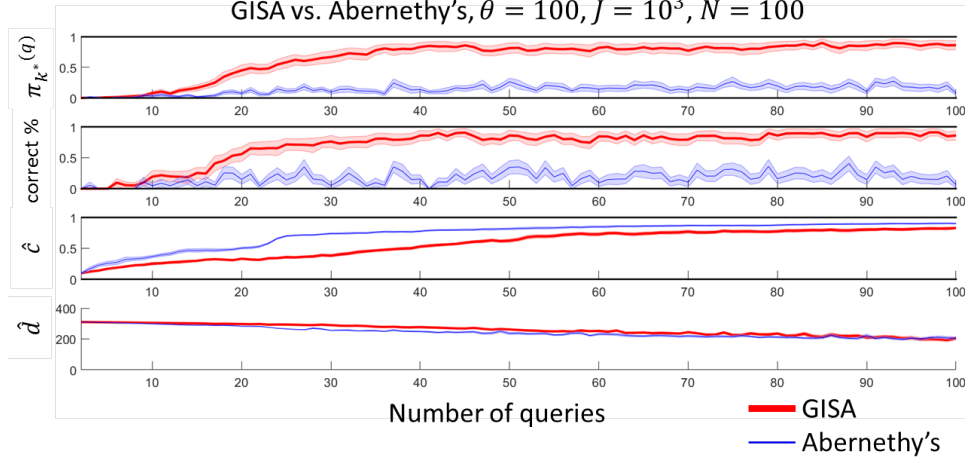


Figure 5: Comparison between GISA and Abernethy’s alg. under $\theta = 100$, $J = 10^3$, and $N = 100$. The standard error of the mean is calculated based on 100 bootstrap samples. Best viewed in color.

We also notice that the norm of $\hat{\mathbf{w}}$ does not converge to that of \mathbf{w}^* in either case. This is because of the plateau in $p(\mathbf{w}|\mathcal{S})$ under the noise-free assumption allows a variety of $\hat{\mathbf{w}}$ to be almost equally likely (see Fig. 4b). This is conceptually the same issue as characterizing the distribution of rare events. Note that $\hat{\mathbf{w}}$ will be close to the origin when response noise exists, and this issue will be alleviated. For completeness, we also note that on average 56% of the queries are chosen from the first half of the heuristic candidate set (based on Π), and the rest from the “minimax variance” criterion. The average initial and final entropy values are $H(\Pi^{(0)}) = 8.22$ and $H(\Pi^{(100)}) = 0.14$. The initial entropy is lower than the theoretical maximum ($\log_2 2455 \approx 11$) since products with higher prices are intrinsically more likely to be optimal under a normal prior.

Effects of algorithmic parameters To further understand the influence of J and N on GISA’s performance, we run the algorithm under $(J = 10^4, N = 100)$ and $(J = 10^3, N = 2)$ with $T = 20$ and $Q = 100$. Based on our GISA implementation, the case with $N = 2$ represents choosing between two candidate queries based on Eq. (8): One has two designs with the highest probabilities, and the other is the choice by Abernethy’s alg.. Fig. 6 summarizes the comparison between $J = 10^4$ and $J = 10^3$. The result shows

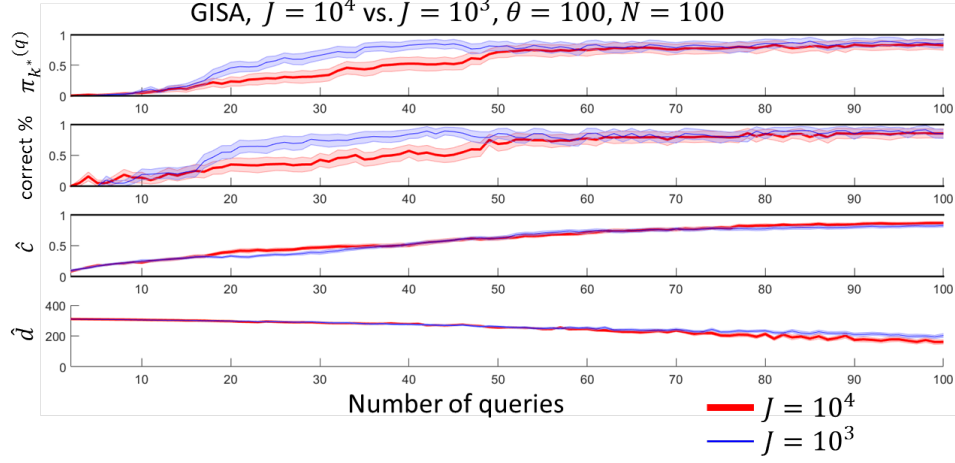


Figure 6: Comparison between $J = 10^4$ and $J = 10^3$ under GISA, $\theta = 100$, and $N = 100$. Best viewed in color.

that an increased MCMC sample size could lead to lower convergence in $\pi_{k^*}^{(q)}$. This is because the large sample set may have a better chance to cover segments, especially the small ones, within the cone. Fig. 7 summarizes the comparison between $N = 100$ and $N = 2$. By investigating results from individual simulations, we find that $N = 100$ significantly outperforms $N = 2$ only when \mathcal{W}_{k^*} is narrow. This indicates that a larger set of queries, i.e., ways to cut the part-worth space, will allow GISA to perform more effectively. For concreteness, we report details of one particular case: Let $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3$ be designs with attribute levels listed in Table 5. Under the settings from Sec. 5, when \mathbf{z}_3 is picked as the single competing product, \mathbf{z}_1 is the true optimal design. Fig. 8b shows that one may identify an incorrect optimal design (\mathbf{z}_2) when the part-worth estimator is slightly deviated from \mathbf{w}^* in the dimensions for which the red lines are close to the black blocks. This shows that \mathcal{W}_{k^*} is indeed narrow in this situation. Fig. 8a compares the performance of GISA with $N = 100$ and $N = 2$ under this particular case, with $T = 20$ and $Q = 200$.

5.5 Influence of response noise (or preference indifference)

We show in Fig. 9a the comparison under $\theta = 1$, $J = 10^3$, and $N = 1000$. The false response rate in this case is 48% for a utility gap of 0.1. The result shows that we can reach a good part-worth estimation, as \hat{c} and \hat{d}

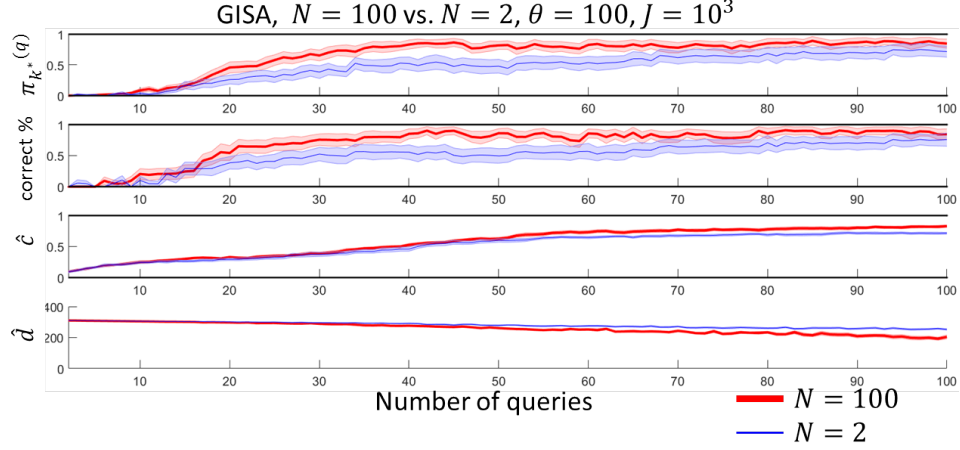


Figure 7: Comparison between $N = 100$ and $N = 2$ under GISA, $\theta = 100$, and $J = 10^4$. Best viewed in color.

Table 5: A case with narrow \mathcal{W}_{k^*}

Design	\mathbf{z}_1	\mathbf{z}_2	\mathbf{z}_3
Weight Capacity	300	300	250
Aspect Ratio	8/8	8/8	7/8
Platform Area	120	120	140
Tick Mark Gap	5/32	5/32	3/32
Number Size	1.25	1.25	0.75
Price	25	30	10

can converge in both GISA and Abernethy’s alg. (which confirms that the algorithms are robust for homogeneous and noisy part-worth estimation), but neither can correctly identify the optimal design. Recall the discussion in Sec. 2.5, the failure here is due to the clustering of segments in the part-worth space close to the origin. One example of this situation is visualized in Fig. 9b, which shows the narrow space of \mathcal{W}_{k^*} around \mathbf{w}^* . Under this situation, and with the high uncertainty of the part-worth estimation due to high noise, the probability mass will be shared among all product candidates corresponding to the segments within the cluster around \mathcal{W}_{k^*} . With an intermediate noise level $\theta = 10$ (false response rate 27% for a utility gap of 0.1), we still observe significant performance drop from GISA. See Fig. 10. Therefore, correct identification of the optimal design under noisy user responses is challenging by nature, as it requires both high accuracy in

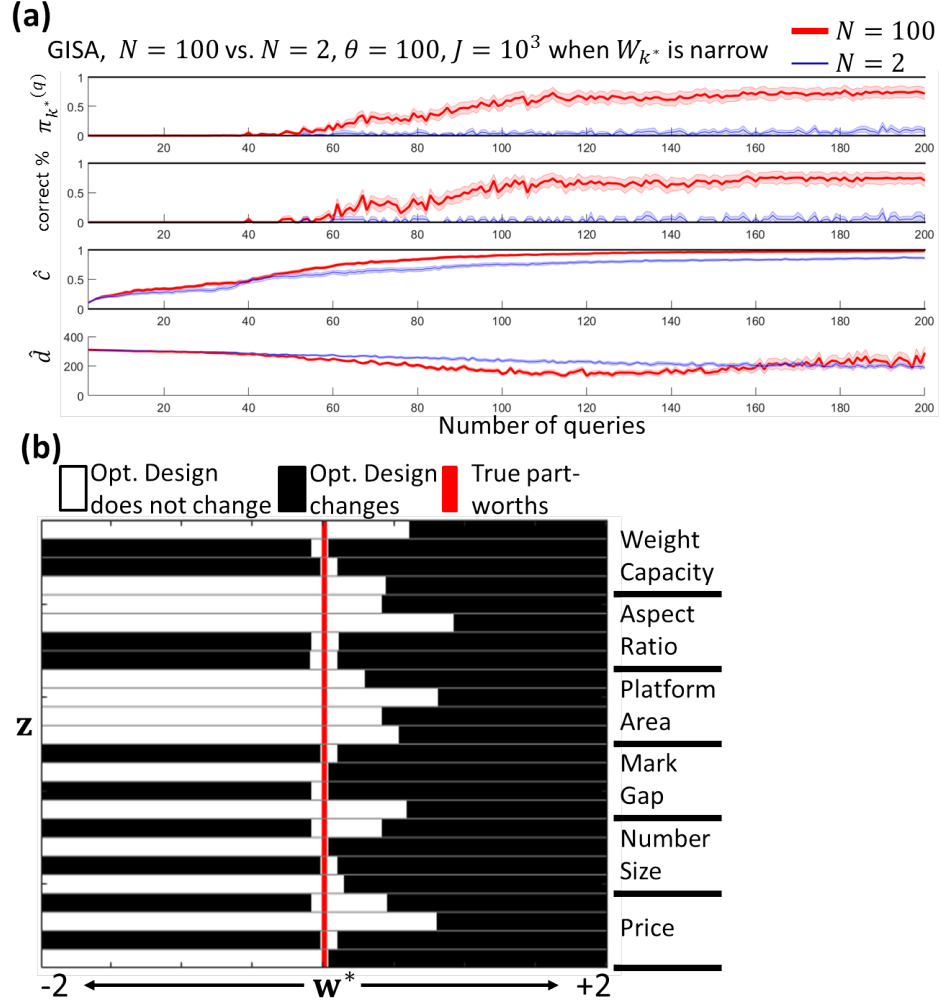


Figure 8: (a) Comparison between GISA performance with $N = 100$ and $N = 2$ under the case in Table 5 ($\theta = 100$, $J = 10^3$). (b) An illustration of how small deviations from \mathbf{w}^* will change the optimal design when \mathcal{W}_{k^*} is narrow: Each row represents part-worth values of an attribute level. Best viewed in color.

the point estimator and low uncertainty in the estimation.

While GISA suffers from noise responses, the comparison among the three noise levels (Fig. 11) shows that the product chosen through GISA consistently has a higher profit than that through Abernethy’s alg., with its advantage decreases with increasing noise. This finding highlights the necessity of GISA, as it shows that for two part-worth estimates close to the ground truth in correlation (and thus they could belong to two neighbouring segments in the part-worth space), the difference between their profits can be significant, especially when noise response is low. We shall also note that the noisy preference causes the profit variance among all products to be less significant, which allows both GISA and Abernethy’s alg. to achieve a profit close to the true optimum when an adequate number of queries are made, even when $\theta = 1$.

6 Discussion

Applicability to real-time survey Fig. 12 records the response time of GISA and Abernethy’s alg. when executed on a workstation with Intel Xeon CPU at 2.10GHz, with a sample size of $J = 10^3$, a candidate query size of $N = 100$, a candidate design size of $K = 2455$, and a total query size of 3012285. The query selection steps for Abernethy’s alg. is slightly slower than that of GISA as our implementation requires sorting of two arrays, both starting with 3012285 elements, while GISA requires one. The cost of numerical integrals in GISA increases linearly with the size of \mathcal{S} . We also note that the computational complexity of estimating Π scales linearly with the number of candidate designs. While GISA could thus become impractical if we consider candidate designs induced by a large set of attributes, such situations rarely happen in questionnaires as human beings can only consider a handful of attributes at a time.

Heterogeneous preferences and alternative preference models Recall that GISA only requires $p(\mathbf{w}; \mathcal{S})$ for choosing a query, thus the algorithm can be extended for heterogeneous user preferences. To do so, let the true individual part-worths be drawn from a distribution $p(\mathbf{w}; \boldsymbol{\beta})$ with unknown parameters $\boldsymbol{\beta}$. For example, $\boldsymbol{\beta}$ can represent the mean and variance-covariance matrix of \mathbf{w} , or hyperparameters of distributions where these two are drawn from. We will then redefine $p(\mathbf{w}; \mathcal{S}) := \mathbb{E}_{\boldsymbol{\beta}} p(\mathbf{w} | \boldsymbol{\beta})$ with respect to the learned posterior $p(\boldsymbol{\beta}; \mathcal{S})$. In addition, other preference models, e.g., probit and non-compensatory models, can also be incorporated, provided that a function

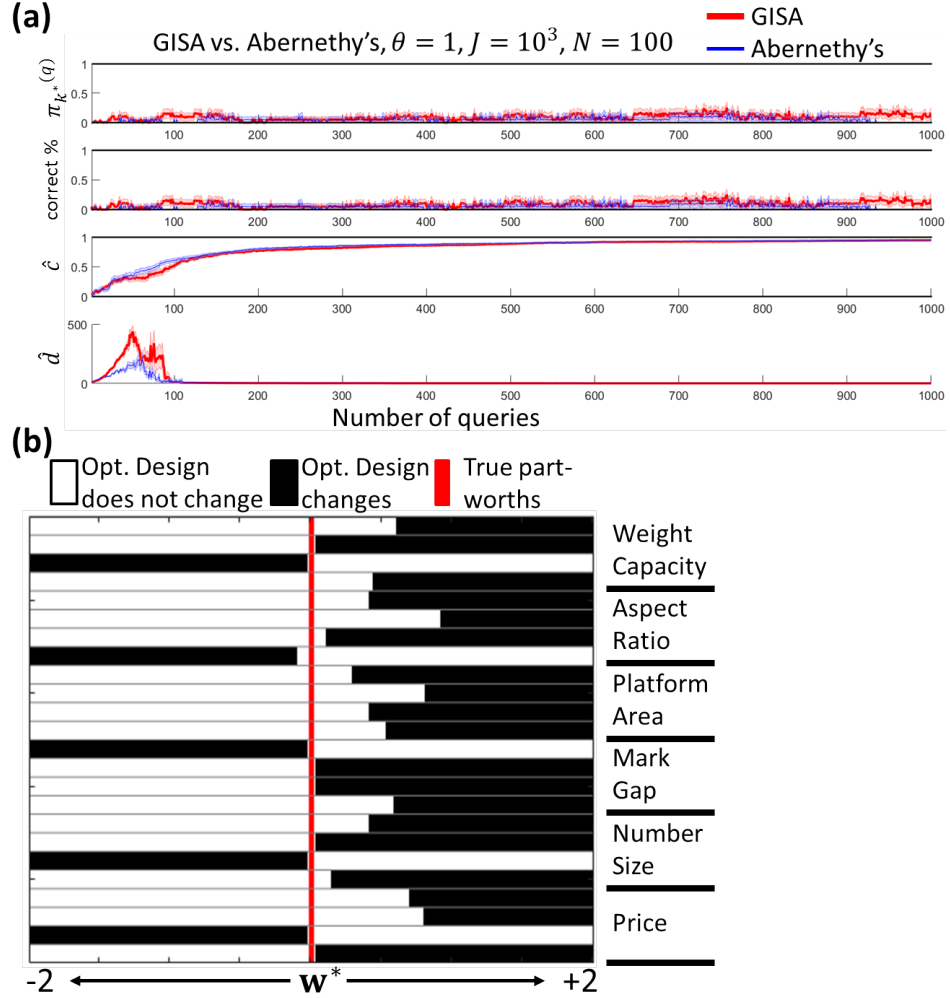


Figure 9: (a) Comparison between GISA and Abernethy's alg. under $\theta = 1$, $J = 10^3$, and $N = 100$. (b) An illustration of how small deviations from \mathbf{w}^* will change the optimal design in the case of $\theta = 1$: Each row represents part-worth values of an attribute level. Best viewed in color.

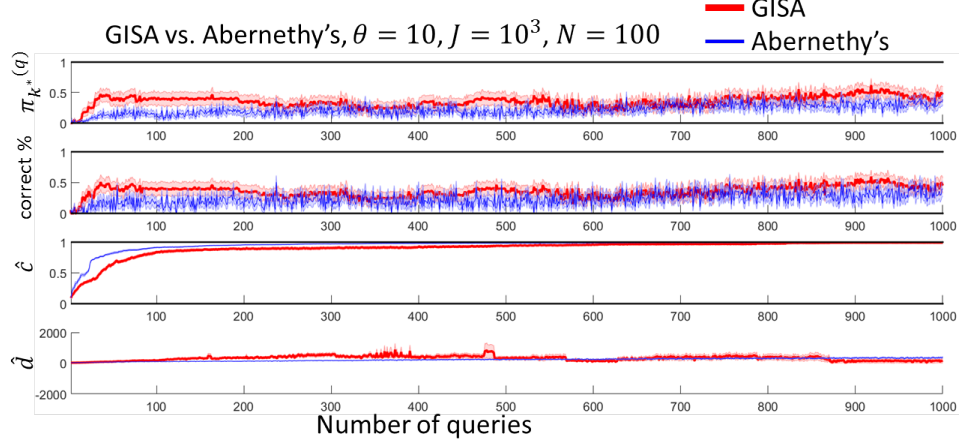


Figure 10: (a) Comparison between GISA and Abernethy’s alg. under $\theta = 10$, $J = 10^3$, and $N = 100$. (b) An illustration of how small deviations from \mathbf{w}^* will change the optimal design in the case of $\theta = 1$: Each row represents part-worth values of an attribute level. Best viewed in color.

proportional to the posterior distribution $p(\mathbf{w}; \mathcal{S})$ can be calculated (\mathbf{w} will be a binary vector in the non-compensatory case [11]). Nonetheless, such extensions may significantly increase the computation cost for calculating $\hat{\mathbf{w}}$ and the numerical integrals involved in GISA, due to the additional integral involved in probit models and the combinatorial nature of estimating \mathbf{w} of a non-compensatory model.

Optimal query Recall that the current implementation picks candidate queries heuristically to make the algorithm tractable. Theoretically, one can minimize the local cost $L^{(q)}$ with respect to the query without limiting feasible solutions to \mathcal{Q} , so that queries can be “designed” rather than chosen from the set. The cost-effectiveness of such an approach is yet to be studied.

Beyond binary queries Our questionnaire assumes no “outside good”, i.e., a user has to make a decision between the two presented designs. This can be relaxed by allowing the user to choose “not to buy any of these” and/or “these are equally good”. The former requires the estimation of the utility of “not choosing any”; and the latter sets a constraint on the utility gap between the two designs. The functional form of the posterior distribution $p(\mathbf{w}|\mathcal{S})$ can be modified to incorporate both responses. In addition, it

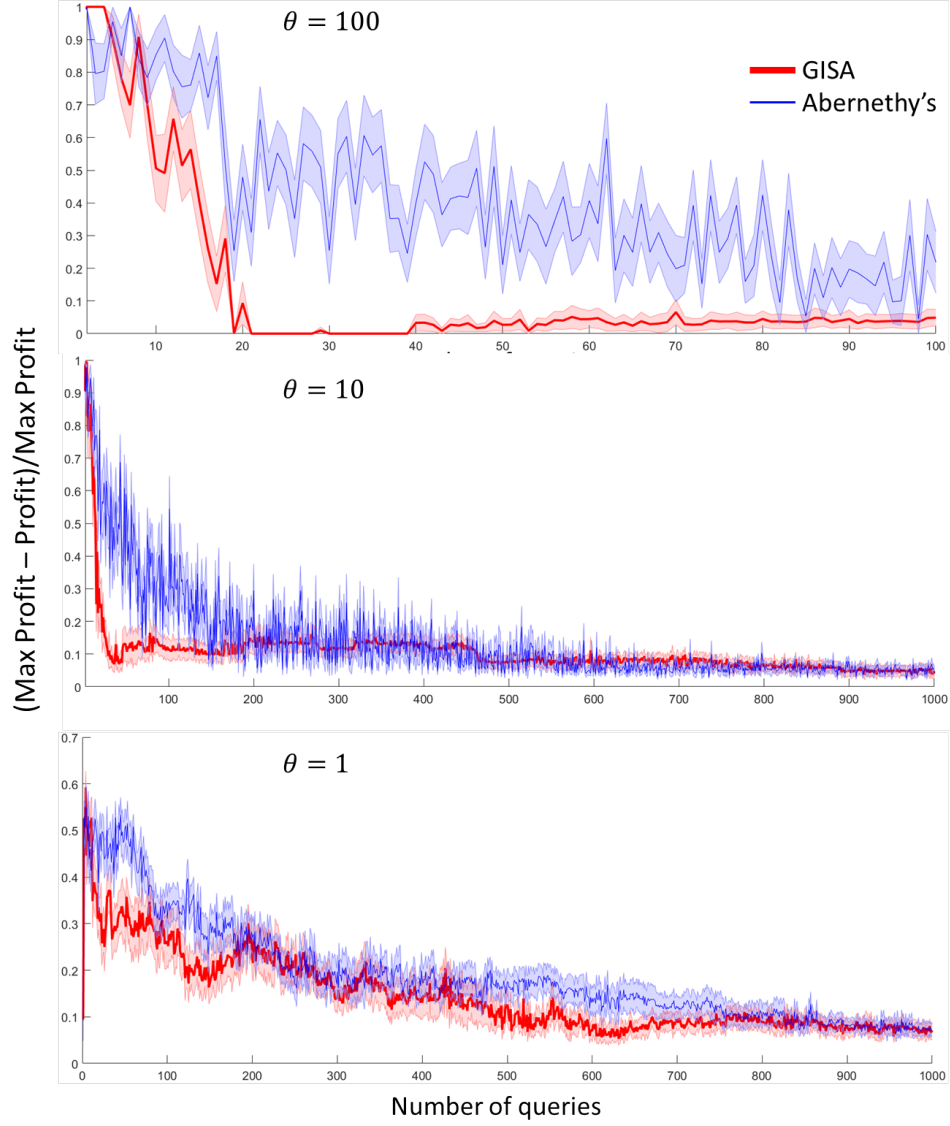


Figure 11: Profit gaps from GISA and Abernethy's alg. under $\theta = 1, 10, 100$, $J = 10^3$. The standard error of the mean is calculated based on 100 bootstrap samples. Best viewed in color.

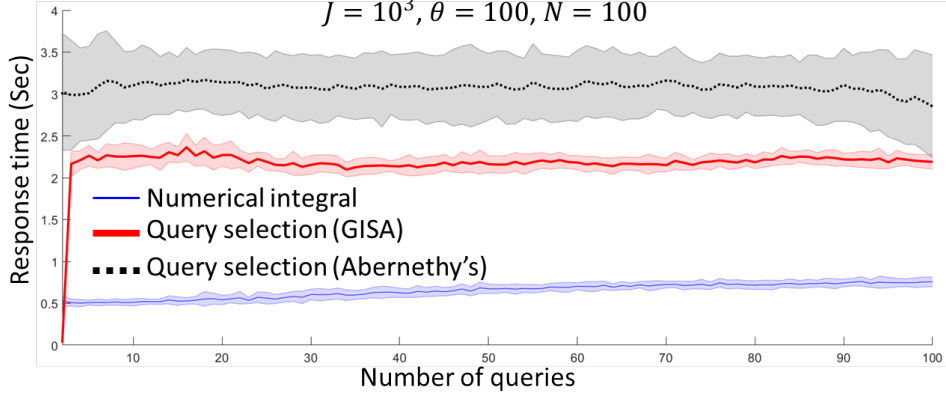


Figure 12: Query response time for GISA and Abernethy’s alg. under the presented case (2455 designs in \mathbb{R}^{24}).

is feasible to extend the query to include more than two designs, in which case the user can choose more than one design as the more preferred ones. Such choice data can be interpreted as multiple binary choices and thus can be directly incorporated into the regularized logistic model. One caveat in doing so is that when random errors in the choice of multiple products during one query is correlated, a probit model will be used in place of multiple logit ones, which makes the estimation much harder.

Product line design While this paper focused on the design of a single product, optimal product line design can be handled by considering each possible combination of products as a meta-design, i.e., during the questionnaire, we will calculate the probability for product lines to be the most profitable, instead of individual designs. The computational cost for query selection increases linearly with the number of candidates (combinations of designs). Therefore with the combinatorial nature of product lines, this problem could still be intractable. Nonetheless, such an issue is universal to product line design, regardless of the choice of a query strategy. The added value of this paper is that it enables a more effective query when one can afford it. We shall also note that the cost of numerical integration will not be affected as the dimensionality of the part-worth space does not change.

Alternative definition of optimality Recall that in Subsec. 2.3 we proposed two definitions of optimality under the existence of uncertainty in

part-worth estimation: The design with the highest probability to be the most profitable, and that with the highest expected profit. These two are different in general: The former may have less expected profit, while the latter may have higher variance in profit. Nonetheless, when we reach $\pi_{k^*} = 1$, the optimal design \mathbf{z}_{k^*} is the solution under both definitions. However, we have not yet investigated which of these two measures will lead to faster correct identification of the optimal product design, and in what conditions.

Infinite number of candidate products While the proposed query method is developed for a finite candidate set, it is sometimes desired that the optimal product can be found from a continuous space. To take dial-readout scale case as an example, once we identified that a price at \$10 is the optimal setting among all five levels, one would like to know if \$9 or \$11 would lead to even better profit. This refined identification problem can be solved by continuing the query process with the current $p(\mathbf{w}|\mathcal{S})$ and a new segmentation of the part-worth space produced by the refined candidate products.

Relevance to value-based global optimization (VGO) VGO [29] is the process of solving black-box optimization problems through an adaptive sampling scheme where each sample is chosen to maximize the “value of information”, i.e., the expected difference between the potential profit after one more sample (minus the cost of the sample) and the current profit. VGO is similar to the proposed method in that it uses a local objective to guide the query. However, the value of information measure is not derived directly from solving Eq. (4) or Eq. (5). In addition, the original formulation of VGO assumes query responses to be evaluations of the objective (e.g., profit), whereas in this paper, we focused on queries that are pair-wise comparisons of a measure (preference) that can indirectly infer the objective. It would be interesting, nonetheless, to extend VGO to the setting of optimal product identification in an extension to this work.

7 Conclusions

While most research on design for market systems has focused on post-data-collection scenarios where adequate data is assumed to be available for identifying the optimal designs, this work investigated how cost-effective questionnaires shall be adaptively constructed for the purpose of identifying the most profitable design from a finite candidate set.

We showed that the two objectives, preference modeling and optimal product identification, require their own questionnaire mechanisms due to their difference in problem formulation. To summarize, the former prefers queries that quickly concentrate the part-worth distribution towards the true part-worths, while the latter requires queries to shift the distribution to a segment of the part-worth space that is defined by the true optimal design. Based on this, we proposed an extension of a group identification algorithm (GISA) that directly minimizes the expected number of queries needed to identify the optimal product. While developed based on a noise-free model, GISA consistently outperformed a standard adaptive questionnaire algorithm with respect to both the accuracy of identifying the true optimal design and the resultant expected profit. This consistency is shown through multiple independent runs of a simulated case study with randomized settings. The advantage of GISA, however, reduces with increasing response noise level. Our findings encourage deeper investigation into knowledge acquisition strategies that smartly seek for and integrate engineering and marketing data to facilitate low-cost and accurate identification of the optimal product design.

Acknowledgement

The authors would like to thank Professor Fred Feinberg for his advice and valuable comments on this work. This work has been supported by the National Science Foundation under Grant No. CMMI-1266184. This support is gratefully acknowledged.

References

- [1] Wassenaar, H. J., and Chen, W., 2003. “An approach to decision-based design with discrete choice analysis for demand modeling”. *Journal of Mechanical Design*, **125**, p. 490.
- [2] Michalek, J. J., Feinberg, F. M., and Papalambros, P. Y., 2005. “Linking marketing and engineering product design decisions via analytical target cascading”. *Journal of Product Innovation Management*, **22**(1), pp. 42–62.
- [3] Wassenaar, H., Chen, W., Cheng, J., and Sudjianto, A., 2005. “Enhancing discrete choice demand modeling for decision-based design”. *Journal of Mechanical Design*, **127**(4), pp. 514–523.

- [4] Lewis, K. E., Chen, W., and Schmidt, L. C., 2006. *Decision making in engineering design*. American Society of Mechanical Engineers.
- [5] Michalek, J. J., Ebbes, P., Adigüzel, F., Feinberg, F. M., and Papalambros, P. Y., 2011. “Enhancing marketing with engineering: Optimal product line design for heterogeneous markets”. *International Journal of Research in Marketing*, **28**(1), pp. 1–12.
- [6] Kang, N., Feinberg, F. M., and Papalambros, P. Y., 2014. “Integrated decision making in electric vehicle and charging station location network design”. In ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference.
- [7] Long, M., and Morrow, W. R., 2015. “Should optimal designers worry about consideration?”. *Journal of Mechanical Design*, **137**(7), p. 071411.
- [8] Shin, J., and Ferguson, S. “Exploring product solution differences due to choice model selection in the presence of noncompensatory decisions with conjunctive screening rules”. *Journal of Mechanical Design*.
- [9] Abernethy, J., Evgeniou, T., Toubia, O., and Vert, J.-P., 2008. “Eliciting consumer preferences using robust adaptive choice questionnaires”. *Knowledge and Data Engineering, IEEE Transactions on*, **20**(2), pp. 145–155.
- [10] McFadden, D., 1973. “Conditional logit analysis of qualitative choice behavior”.
- [11] Hauser, J. R., Toubia, O., Evgeniou, T., Befurt, R., and Dzyabura, D., 2010. “Disjunctions of conjunctions, cognitive simplicity, and consideration sets”. *Journal of Marketing Research*, **47**(3), pp. 485–496.
- [12] Morrow, W. R., Long, M., and MacDonald, E. F., 2014. “Market-system design optimization with consider-then-choose models”. *Journal of Mechanical Design*, **136**(3), p. 031003.
- [13] Shin, J., and Ferguson, S., 2015. “Modeling noncompensatory choices with a compensatory model for a product design search”. In ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, pp. V02AT03A036–V02AT03A036.

- [14] Lenk, P. J., DeSarbo, W. S., Green, P. E., and Young, M. R., 1996. “Hierarchical Bayes conjoint analysis: Recovery of partworth heterogeneity from reduced experimental designs”. *Marketing Science*, **15**(2), pp. 173–191.
- [15] Rossi, P. E., and Allenby, G. M., 2003. “Bayesian statistics and marketing”. *Marketing Science*, **22**(3), pp. 304–328.
- [16] Toubia, O., Evgeniou, T., and Hauser, J., 2007. “Optimization-based and machine-learning methods for conjoint analysis: Estimation and question design”. *Conjoint Measurement*, p. 231.
- [17] Toubia, O., Hauser, J., and Simester, D., 2004. “Polyhedral methods for adaptive choice-based conjoint analysis”. *Journal of Marketing Research*, **41**(1), pp. 116–131.
- [18] Toubia, O., Simester, D., Hauser, J., and Dahan, E., 2003. “Fast polyhedral adaptive conjoint estimation”. *Marketing Science*, **22**(3), pp. 273–303.
- [19] Salisbury, L. C., and Feinberg, F. M., 2010. “Alleviating the constant stochastic variance assumption in decision research: Theory, measurement, and experimental test”. *Marketing Science*, **29**(1), pp. 1–17.
- [20] Louviere, J., Street, D., Carson, R., Ainslie, A., Deshazo, J., Cameron, T., Hensher, D., Kohn, R., and Marley, T., 2002. “Dissecting the random component of utility”. *Marketing Letters*, **13**(3), pp. 177–193.
- [21] Tong, S., and Koller, D., 2002. “Support vector machine active learning with applications to text classification”. *The Journal of Machine Learning Research*, **2**, pp. 45–66.
- [22] Jamieson, K. G., and Nowak, R., 2011. “Active ranking using pairwise comparisons”. In *Advances in Neural Information Processing Systems*, pp. 2240–2248.
- [23] Bellala, G., Bhavnani, S., and Scott, C., 2012. “Group-based active query selection for rapid diagnosis in time-critical situations”. *IEEE Transactions on Information Theory*, **58**(1), pp. 459–478.
- [24] Hyafil, L., and Rivest, R., 1976. “Constructing optimal binary decision trees is NP-complete”. *Information Processing Letters*, **5**(1), pp. 15–17.

- [25] Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J., 2008. “LIBLINEAR: A library for large linear classification”. *The Journal of Machine Learning Research*, **9**, pp. 1871–1874.
- [26] Chib, S., and Greenberg, E., 1995. “Understanding the metropolis-hastings algorithm”. *The american statistician*, **49**(4), pp. 327–335.
- [27] Haario, H., Saksman, E., and Tamminen, J., 2001. “An adaptive metropolis algorithm”. *Bernoulli*, pp. 223–242.
- [28] Boneh, A., and Golan, A., 1979. “Constraints’ redundancy and feasible region boundedness by random feasible point generator (RFPG)”. In Third European congress on operations research (EURO III), Amsterdam.
- [29] Moore, R. A., Romero, D. A., and Paredis, C. J., 2014. “Value-based global optimization”. *Journal of Mechanical Design*, **136**(4), p. 041003.